

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Komunikační zařízení pro modifikaci programu
systému Microcon
Interface for Program Modification of Microcon
System

2014

Matěj Franc

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Zadání bakalářské práce

Student:

Matěj Franc

Studijní program:

B2649 Elektrotechnika

Studijní obor:

2602R014 Aplikovaná a komerční elektronika

Téma:

Komunikační zařízení pro modifikaci programu systému Microcon
Interface for Program Modification of Microcon System

Zásady pro vypracování:

1. Proveďte rozbor řídicích systémů určených pro řízení krokových motorků MICROCON
2. Navrhněte a realizujte zadávací jednotku pro úpravu parametrů řídicího systému MICROCON
3. Na základě pokynů vedoucího bakalářské práce ověřte činnost jednotky v reálné aplikaci

Seznam doporučené odborné literatury:

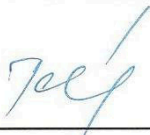
Dle pokynů vedoucího závěrečné práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Petr Palacký, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Ing. Petr Palacký, Ph.D.
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

V Ostravě dne 7.5.2014

Podpis 

Poděkování

Děkuji panu Doc. Ing. Petru Palackému, Ph.D. za poskytnutý čas a rady při zpracování mé bakalářské práce.

Abstrakt

Cílem této práce je návrh zařízení, které bude sloužit k modifikaci uživatelského povelového souboru řídicího systému MICROCON bez potřeby připojení PC. Toto zařízení bude využívat LCD displeje a převodník úrovně MAX232. Samotný text bakalářské práce obsahuje v První kapitole popis použitého mikrokontroléru Atmel, popis řídicího systému MICROCON, popis použitých periférií a popis sériové komunikace. Druhá kapitola je zaměřena na samotný popis a postup zapojení. Třetí kapitola se zabývá programem pro mikrokontrolér ATmega, který je napsán v jazyku C.

Abstract

The aim of this work is to design a device that will be used to modify the user command file of management system MICROCON without PC. This device will use the LCD display and the MAX232 level converter. The actual text of the paper contains the first chapter, a description of Atmel microcontroller, a description of the control systems MICROCON, a description of the periphery and the description of the serial communication. The second chapter focuses on the description itself and the process diagram. A third chapter deals with the program for ATmega microcontroller, which is written in C.

Klíčová slova

Microcon, mikrokontrolér Amel, LCD displej, RS232, jazyk C

Key words

Microcon, Atmel microcontroller, LCD display, RS232, C programming language

Seznam použitých zkratek a symbolů

V	Volt, jednotka elektrického napětí
A	Ampér, jednotka elektrického proudu
DPS	Deska Plošného Spoje
Obr.	Obrázek
LCD	Displej s tekutými krystaly
PC	Počítač
EEPROM	Elektricky mazatelná paměť
LED	Svítivá dioda
PLC	Programovatelný logický automat
W	Watt, jednotka elektrického výkonu
DIP	Pouzdro s vývody ve dvou řadách
ASCII	Kódovací tabulka, která definuje znaky anglické abecedy
USART	Synchronně/Asynchronní sériové komunikační rozhraní
RISC	Architektura mikropočítače s redukovanou instrukční sadou
JTAG	Standart na testování plošných spojů a programování flash paměti
SPI	Sériové rozhraní pro komunikaci s periferiemi
ADC	Převodník analogového na digitální signál
PWM	Pulzně-šířková modulace
SRAM	Statická polovodičová paměť s přímým přístupem
USB	Univerzální sériová sběrnice
GND	Napájecí, signálová zem
MSB	Bit s nejvyšší vahou
LSB	Bit s nejnižší vahou
TTL	Tranzistorově-tranzistorová logika
Atmel	Výrobce mikropočítačů
ATmega	Typ mikropočítače
RGB	Červeno-Zeleno-Modrá (tříbarevná)
atd.	A tak dále
č.	Číslo
H	Henry, jednotka indukčnosti
CR	Speciální netisknutelný znak, který posune kurzor na následující řádek
s	Sekunda, jednotka času
bit	Základní a nejmenší jednotka digitální informace
Hz	Hertz, jednotka frekvence
OTP	Verze, u kterých lze pouze jednou nahrát SW (pak se sami uzamknou)
SW	SoftWare
HW	HardWare
tzv.	Tak zvané
ID	Identifikační
Bios	Implementace základních vstupně/výstupní funkcí při inicializaci HW
HMI	Human-Machine interface = interface, kterým ovládá uživatel zařízení
Baud	Jednotka modulační rychlosti
Log. 0	Nízká logická úroveň
Log. 1	Vysoká logická úroveň
Integer	Celočíselný datový typ v jazyce C
String	Textový řetězec (datový typ) v jazyce C
FLASH	Paměť mikropočítače

Obsah

1	Úvod	8
2	Popis použitých obvodu, zařízení a zapojení	9
2.1	Řídicí systémy MICROCON	9
2.1.1	Programovatelná jednotka (kontrolér) M1486E1	9
2.1.2	Přehled základní řady obvodů M1486	10
2.1.3	Deska pro řízení krokových motorů CD30M	11
2.1.4	Inmotion PC Utilities	12
2.1.5	Povelový soubor a jeho příklad	12
2.2	Mikropočítač ATmega644-20PU	13
2.2.1	Popis mikropočítače ATmega644-20PU	13
2.2.2	AVR Studio, programátor ISP a PROGISP	14
2.3	Periferie připojené k mikropočítači ATmega644-20PU	15
2.3.1	LCD displej WH1602A	15
2.3.2	Sériová linka RS232 a převodník MAX232	16
2.3.3	Ostatní	17
3	Popis, postup a vývoj zapojení	19
3.1	Blokové schéma zapojení	19
3.2	Popis zapojení	20
4	Popis programu mikropočítače	26
4.1	Blokový diagram programu	26
4.2	Příklad modifikace programu Microcon	27
4.2.1	Odeslání programu Microcon z PC do mikropočítače	28
4.2.2	Odeslání programu Microcon mikropočítačem do jednotky Microcon	31
4.3	Popis některých podprogramů	34
5	Závěr	35
	Literatura	36
	Seznam příloh	37

1 ÚVOD

Při výběru tématu bakalářské práce mé rozhodnutí bylo ovlivněno tím, že už delší dobu pracuji ve firmě zabývající se výrobou a kontrolou mikrofiltrů. Jelikož jsem v oddělení výzkumu a vývoje, dostal jsem nápad na vylepšení řídicích jednotek krokových motorů. Takže pokud je bakalářská práce o řešení například nápravy problému, zvolil jsem toto téma, které vyplynulo z praxe.

Řídicí jednotky Microcon se používají k řízení krokových motorů. Uživatel vytvoří v PC v aplikaci Inmotion PC Utilities svůj uživatelský program a může ho editovat a nahrávat do řídicích jednotek Microcon přes sériový port RS232. V uživatelském programu je možnost ovládat kupříkladu: rychlost, směr, akceleraci, otočení o určitý počet kroků, atd. Rovněž lze ovlivňovat chod uživatelského programu změnou binárních vstupů nebo je možno přidávat časové zpoždění. Pokud chce uživatel změnit tyto parametry, musí pomocí PC nahrát opět upravený uživatelský program. V praxi to znamená například chodit s notebookem po provozu výrobní linky. Má bakalářská práce se tento problém snaží zjednodušit.

Cílem bylo vyvinout zařízení, které bude uživatelský program pro řídicí jednotky Microcon upravovat a následně odesílat do těchto řídicích jednotek. Toto zařízení bude umístěno u řídicí jednotky Microcon. Součástí tohoto zařízení bude LCD displej a tlačítka, pomocí kterých uživatel upraví uživatelský program a odešle ho do řídicí jednotky Microcon.

Prvně uživatel pošle přes sériovou linku uživatelský program do mnou vytvořeného zařízení. Toto prvotní naprogramování proběhne pouze jednou, poté bude uživatelský program uložen v zařízení v paměti EEPROM. Vybrané parametry bude možno upravovat a po úpravě možno odeslat do řídicí jednotky Microcon.

V první části jsou popsány použité obvody, zařízení a zapojení. Další část se zabývá popisem a vývojem zapojení, která byla aplikována. Poslední část bakalářské práce je zaměřena na program použitého mikropočítače ATmega.

2 POPIS POUŽITÝCH OBVODU, ZAŘÍZENÍ A ZAPOJENÍ

2.1 Řídicí systémy MICROCON

Základní rozdělení produktu MICROCON by bylo na programovatelné jednotky a desky pro řízení krokových motorů. Programovatelné jednotky lze přirovnat k malému programovatelnému automatu, tedy něco jako PLC. Tyto jednotky mají sériovou linku, DA převodníky, binární vstupy a výstupy a svoji paměť. Desky pro řízení krokových motorů slouží k propojení programovatelné jednotky Microcon a výkonového stupně, který je součástí této desky (na této desce jsou i konektory, tlačítko, jumpery, ...). Výkonové stupně (zesilovače) jsou řízeny řídicím signálem z programovatelné jednotky nebo PLC. Napájí samotný krokový motor. V této části bude podrobnější rozbor použité programovatelné jednotky, desky SD30 a programu Inmotion PC Utilities pro vytváření a editování povelového souboru.

2.1.1 Programovatelná jednotka (kontrolér) M1486E1

Tato jednotka je vlastně mikrokontrolérem, který používá svůj povelový soubor. Umožňuje realizovat samostatné řízení připojeného zařízení nebo stroje. To znamená, že se k této jednotce připojí krokový motor a vstupně/výstupní logické signály. Jednotka M1486E1 posílá řídicí signály pro externí výkonový zesilovač (v našem případě deska CD30M), napájející krokový motor.

Základní parametry jednotky M1486E1 jsou například:

- EEPROM paměť o velikosti 2000 bitů
- 21 univerzálních vstupů/výstupů
- Mají sériový kanál, na kterém může být připojeno až 16 těchto jednotek
- Nastavitelnost rychlosti až do 40000 kroku/s
- Až 64 mikrokroků na jeden celý krok
- Délka dráhy pohybu až 16 milionů kroků
- Jednotkou lze ovládat plynulé zrychlení i brzdění

Elektrické vlastnosti jednotky M1486E1:

- Napájecí napětí minimálně 4,5 V a maximálně 5,5 V
- Přivedené vstupní napětí jednotlivých vývodů vůči zemi je bezpečné od -0,5 V do 6,5 V
- Maximální vstupní nebo výstupní proud jednotlivých vývodů je do 10 mA
- Maximální ztrátový výkon na jednotce může být do 200 mW

B8/D1.0	1	40	Vcc
B9/D1.1	2	39	Limit
DUAL/D1.2	3	38	B1
S&D/D1.3	4	37	B2
ST/D1.4	5	36	B3
Dir/D1.5	6	35	B4
D1.6	7	34	B5/BR0
PHASE1	8	33	B6/BR1
RST	9	32	B7/JD
RDX	10	31	Vcc
TDX	11	30	NC
BOOST/JOG	12	29	NC
SD	13	28	B17/PHASE2
B18/ADDR0	14	27	B16/D2.6
B19/ADDR1	15	26	B15/D2.5
B20/ADDR2	16	25	B14/D2.4
B21/ADDR3	17	24	B13/D2.3
XTAL2	18	23	B12/D2.2
XTAL	19	22	B11/D2.1
GND	20	21	B10/D2.0

Obr. 1 Zapojení vývodů programovatelné jednotky M1486E1 (převzato z [2])

Na Obr. 1 je zapojení vývodu jednotky M1486E1. Funkce vývodů: napájení, univerzální vstupy/výstupy, reset, sériová linka, vstup pro připojení krystalu, vstupy nastavující rychlost sériové linky, výstupy pro řízení směru pohybu, vstupy pro řízení krokování pohybu, výstupy určující směry proudu ve fázích motoru, vstupy pro nastavení proudů ve fázích motoru, vstupy určující adresu kontroléru (v případě připojení více kontrolérů na jednu komunikační linku), vstup ovládající režim snížené spotřeby a vstupy nastavující módy řízení.

2.1.2 Přehled základní řady obvodů M1486

Již popsaná programovatelná jednotka M1486E1 je z řady jednotek M1486. V této části je napsán rozbor této řady jednotek. Řada se rozděluje na dva typy. Výběr typu jednotky je podle systému a funkce, kam má být aplikována.

První skupina jednotek používá paměť povelového souboru typu RAM. Toto znamená, že po vypnutí napájení dojde k vymazání povelového souboru z jednotky. Po zapnutí je nutné do jednotky nahrát povelový soubor. Proto se jednotky tohoto typu používají v aplikacích s nadřazeným systémem (PLC, PC). Následující rozdělení je z [2].

- M1486A – Paměť RAM má kapacitu 864 bitů a rychlost lze provozovat maximálně do 20 000 kroků/s (při použití krystalu 24 MHz lze zvýšit rychlost až na 40 000 kroků/s)
- M1486A2x – Je zcela shodná s M1486A ale při připojení krystalu 24 MHz je veškeré činnost jednotky zrychlena na dvojnásobek
- M1486B – Plně kompatibilní s M1486A ale s větší kapacitou paměti RAM pro povelů 2032 bitů
- M1486B-NV – Tato jednotka má speciálně přidanou funkci „změna rychlosti za běhu motoru“ a je plně kompatibilní s M1486B

Do druhé skupiny se řadí jednotky, které místo paměti RAM mají paměť EEPROM. Díky této EEPROM paměti zůstanou povely uloženy v jednotce i po vypnutí napájení. V případě, že budeme potřebovat změnit povelový soubor, stačí se připojit a počítačem nový soubor odeslat.

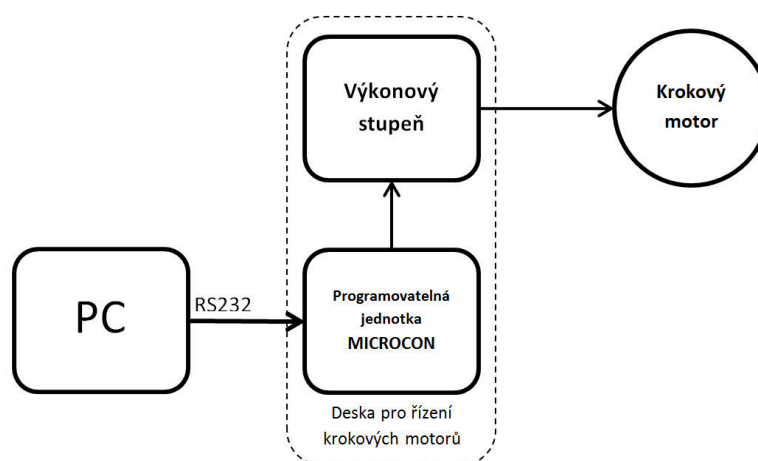
- M1486E1 – Velikost EEPROM paměti je 2000 bitů. Když se jednotka pomocí HW resetu restartuje, program se začne provádět od začátku
- M1486E1-L4 – OTP verze (OTP = program lze do jednotky nahrát pouze jednou, po nahrání se jednotka uzamkne) jednotky M1486E1
- M1486E4 – Kompatibilní s M1486E1, jen s větší kapacitou EEPROM paměti 8000 bitů.

2.1.3 Deska pro řízení krokových motorů CD30M

CD30M je deska, na které jsou komponenty pro pohon krokového motoru. Od řídicí části (vložená programovatelná jednotka M1486) až po výkonový zesilovač s pulsní regulací proudu s možností mikrokrokování. Na desce se nachází DIP přepínače pro nastavení funkce vstupů a výstupů a pro nastavení amplitudy výstupního proudu (0,4 A až 3,3 A). Dále se na desce nachází tlačítko resetu jednotky, dva konektory pro připojení linky RS232 (druhý je použit, pokud je na linku připojeno více těchto desek), uživatelský konektor pro vstupně/výstupní signály a jumpery, které mají funkci přivedení galvanické země z optočlenových vstupů k jednotce.

Parametry koncového stupně [4]:

- Maximální stejnosměrné napájecí napětí 48 V
- Maximální amplituda výstupního proudu 3,3 A
- Minimální indukčnost vinutí připojeného motoru 2 mH



Obr. 2 Blokové schéma zařízení před aplikací bakalářské práce

2.1.4 Inmotion PC Utilities

Uživatel vytváří, edituje a ověřuje uživatelský program v aplikaci Inmotion PC Utilities. Rovněž umožňuje vyslání jednotlivých povelů nebo celého uživatelského povelového souboru do řídicí jednotky Microcon. Součástí tohoto programu je výpočtový modul Motion Calculator (Kalkulátor pohybů), který slouží k výpočtu parametrů pohybu motoru dle zadaných požadavků. Samotný uživatelský program se skládá z ASCII znaků. Na Obr. 16 je zobrazeno prostředí programu při editování uživatelského programu.

2.1.5 Povelový soubor a jeho příklad

Kontrolér používá svůj vlastní povelový soubor. Těchto 50 povelů umožňuje větvení programu dle stavu vstupních signálů nebo dle stavu vnitřních proměnných. Jak již bylo zmíněno, povel se skládá z ASCII znaků. Je možné použít jak malé tak velké písmena. Struktura každého řádku je složena takto: Povel, Argument a Oddělovací znak. Povel je většinou písmeno nebo znak, argument je většinou číslo a oddělovací znak je po stisku klávesy Enter (další řádek) v hexadecimální soustavě „0D“ (netisknutelný znak). Přehled povelů je v příloze [B].

Příklad uživatelského povelového souboru:

x0	// Index – značí, že je tento povelový soubor pro jednotku č. 0 (používá se, //pokud je více jednotek na sériové lince)
[// Disable – začátek podprogramu
S200	// Start/Stop – počáteční rychlost = 200 kroků/s
V1000	// Velocity – maximální rychlost = 1000 kroků/s
A5000	// Acceleration – zrychlení 5000 kroků/s ²
F39616	// Forward – zadaná dráha 39616 mikrokroků dopředu
L3	// Loop – následující povel budou provedeny třikrát
R	// Run – povel pro zahájení vykonávání pohybu
W500	// Wait – zpoždění (čekej) 250 milisekund
E	// End of loop – konec smyčky Loop
]	// Enable – konec podprogramu

2.2 Mikropočítač ATmega644-20PU

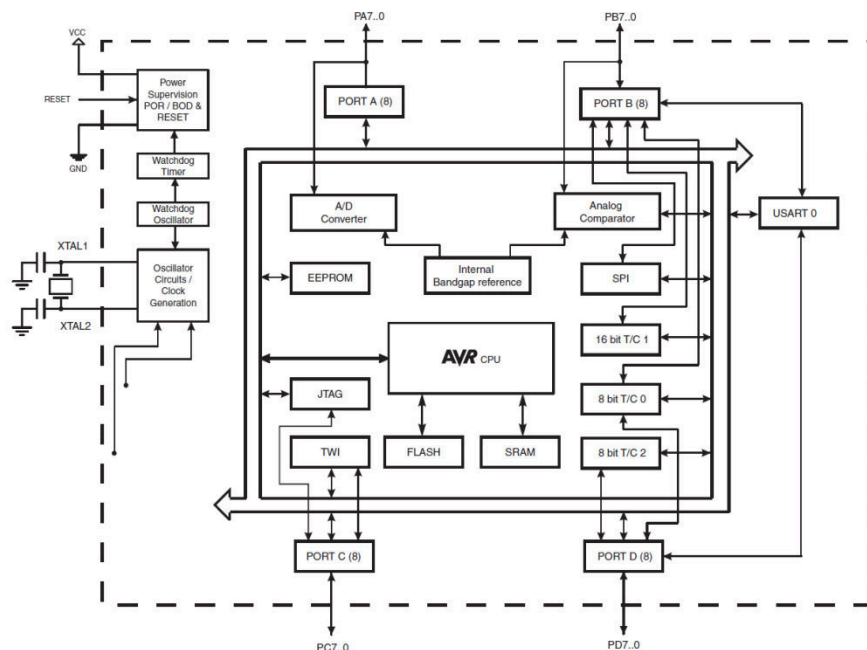
V době výběru mikropočítače na tuto bakalářskou práci jsem nevěděl, kolik budu přesně potřebovat vstupů/výstupů a jaké periferie bude muset mikropočítač obsahovat. Parametry výběru, které musel mikropočítač splňovat, byly například:

- Větší počet vstupů/výstupů
- USART
- Větší paměť programu FLASH a paměť EEPROM

Rozhodl jsem se pro mikropočítač firmy Atmel, jelikož jsem s ním ještě nepracoval a chtěl jsem zkusit něco nového. Po delším rozhodování byl vybrán mikropočítač ATmega644-20PU.

2.2.1 Popis mikropočítače ATmega644-20PU

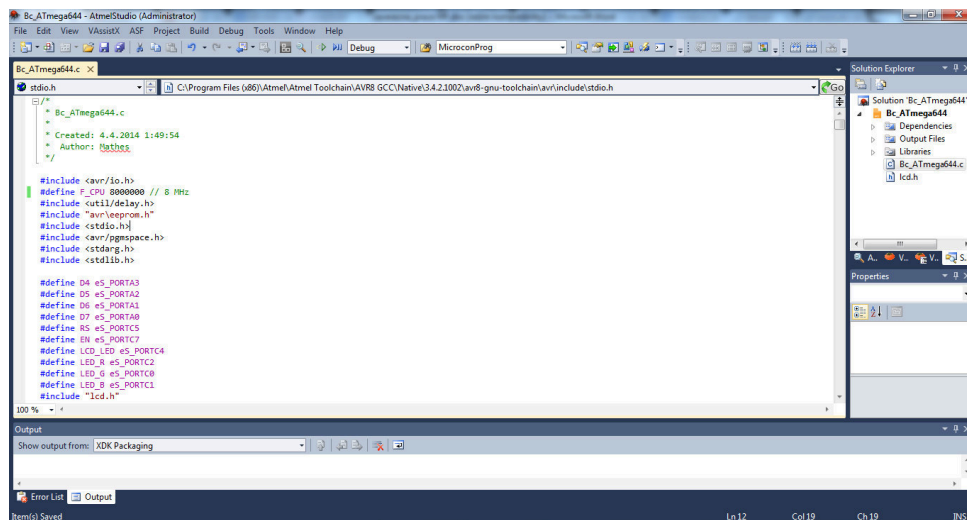
Tento 8mi bitový mikropočítač je architektury RISC. Pracovní frekvence je možná až do 20 MHz. Z hlediska paměťového obsahuje 64kB ISP flash paměť, 2kB EEPROM paměť, 4kB paměť SRAM. Jako vstupně-výstupní brány lze využít PORTA, PORTB, PORTC a PORTD. Jako hlavní periferie bych uvedl: dva 8bitové čítače/časovače, jeden 16bitový čítač/časovač, hodiny reálného času, PWM výstupní kanály, 10bitový ADC, programovatelný USART, SPI, JTAG a mnoho dalších. Mikropočítač je možno napájet od 3 V do 5 V. Jeho vnitřní blokový diagram je na Obr. 3. Pouzdro mikropočítače bylo zvoleno DIP (DIL) pro jednodušší práci a možnost testování na nepájivém poli.



Obr. 3 Blokový diagram mikropočítače ATmega644 (převzato z [1])

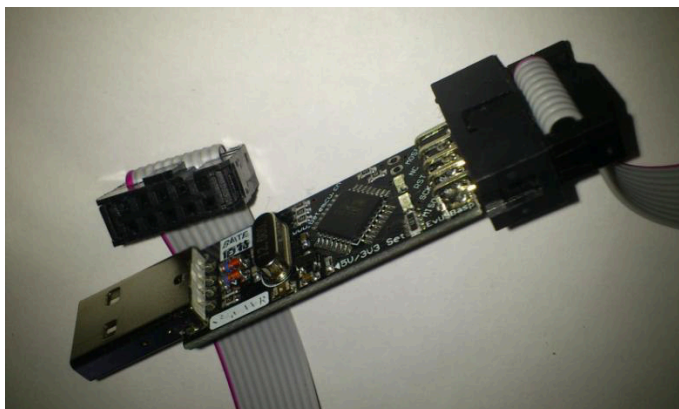
2.2.2 AVR Studio, programátor ISP a PROGISP

Vytvoření a vývoj firmwaru a softwaru probíhalo v prostředí programu Atmel Studio 6 (Obr. 4). Při zakládání nového projektu se vybere typ mikropočítače a poté se už může psát samotný kód. Přes tento program lze psát kódy i pro mikroprocesory Atmel typu AVR, ARM a Cortex. Před nahráním programu do mikropočítače se musí vždy kód zkompilevat. Takto se vytvoří hex soubor a tento pak nahrajeme do mikropočítače.



Obr. 4 Prostředí programu Atmel Studio 6

K programování mikropočítače ATmega644 byl používán programátor AVRISP mkII (Obr. 5). Před použitím bylo pouze nutné nainstalovat ovladače, jelikož se připojuje k PC přes USB. Z USB je rovněž i napájen. K mikropočítači se připojuje pomocí 2x5ti pinového konektoru. Tento konektor je k mikropočítači připojen na nožičkách MOSI, MISO, RESET a GND.



Obr. 5 Programátor AVRISP mkII

K nahrávání HEX souboru do mikropočítače bylo zapotřebí použít jiný program než Atmel Studio, a to, PROGISP (Obr 12). V PROGISP jsem přenastavil tzv. Fuse bity tak abych zablokoval JTAG a zapnul děličku 8mi. Poté stačí načíst HEX soubor a naprogramovat do mikropočítače.

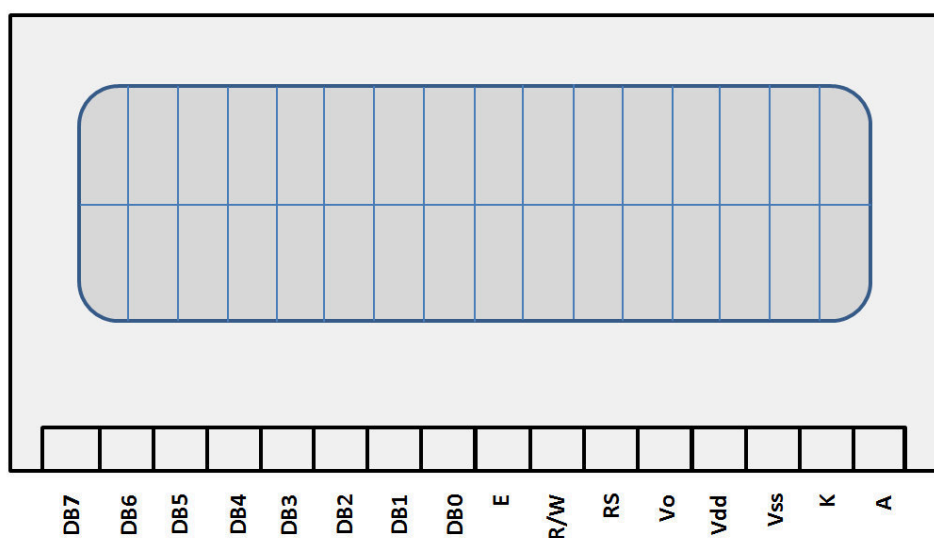
2.3 Periferie připojené k mikropočítači ATmega644-20PU

V této části popíšu připojené periferie k mikropočítači ATmega644. Jedná se především o LCD displej, převodník úrovně MAX232, samotnou sériovou linku RS232, indikační LED, zapojení tlačítek na vstupy mikropočítač.

2.3.1 LCD displej WH1602A

Aby uživatel viděl, co upravuje, bylo potřeba použít nějaké zobrazovací zařízení. Nejideálnější bylo použití LCD displeje (datasheet dostupný v [9]). Tento displej může zobrazit 32 znaků a to v 16ti sloupcích a dvou řádcích. Každý znakový segment se skládá z 7mi pixelů na výšku a 5ti pixelů na šířku. Barva znaků je bílá a barva podsvícení je modrá. Zobrazované znaky jsou znaky z ASCII tabulky. Displej je napájen +5 V, které jsou přivedeny na pin Vdd. Na pinu Vss je připojena zem (GND). V datasheetu je napsáno, že může být napájení i pouze +3 V. Displej je vybaven svým vlastním řadičem s pamětí. Používají se i displeje připojené k mikropočítači po sériové lince ale tento použitý typ je paralelní.

Pokud chceme displeji poslat nějaký komand nebo data, nastavujeme datové vstupy displeje DB0 až DB7. Vstup E je Enable, při jeho nastavení dochází k povolení DB0 až DB7 vstupů do displeje. Pokud je na pinu R/W log. 1 (= Read) tak lze z displeje číst znaky uložené v paměti displeje. Pokud je ale na pinu R/W log. 0 (=Write) tak se data do paměti displeje zapisují. Vstupem RS se říká displeji, zda vstupní data budou komandy nebo data. Dále je na displeji analogový vstup Vo, kterým se nastavuje kontrast displeje. Nastavování je vyřešeno odporovým trimrem, který je jezdcem připojen na tento pin Vo. Analogová hodnota napětí, nastavení kontrastu, se pohybuje mezi Vdd a Vss (+5 V až 0 V). Poslední dva piny jsou využity pro LED podsvícení celého LCD displeje. Pin „K“ je Katoda LED podsvícení a pin „A“ je Anoda. Podsvícení je napájeno cca 4,2 V a odebírá proud do 100 mA. Na Obr. 6 jsou zobrazeny vstupy a výstupy displeje.



Obr. 6 Vstupy/Výstupy použitého displeje

Jak bylo už zmíněno, do displeje se posílají buď data, nebo tzv. komandy. Komandy jsou příkazy, instrukce pro displej, kterým ho nastavujeme a řídíme. Mezi tyto komandy například patří:

- CLEAR – Vymazání všech znaku z displeje
- RETURN HOME – Kurzor se vrátí na základní pozici
- ENTRY MODE SET – Nastavení směru pohybu kurzoru
- DISPLAY ON/OFF CONTROL – Vypnutí displeje nebo kurzoru nebo blikání znaku v místě kde se aktuálně kurzor nachází
- CURSOR & DISPLAY SHIFT – Posune kurzor a displej bez toho aby měnil obsah znakové paměti displeje

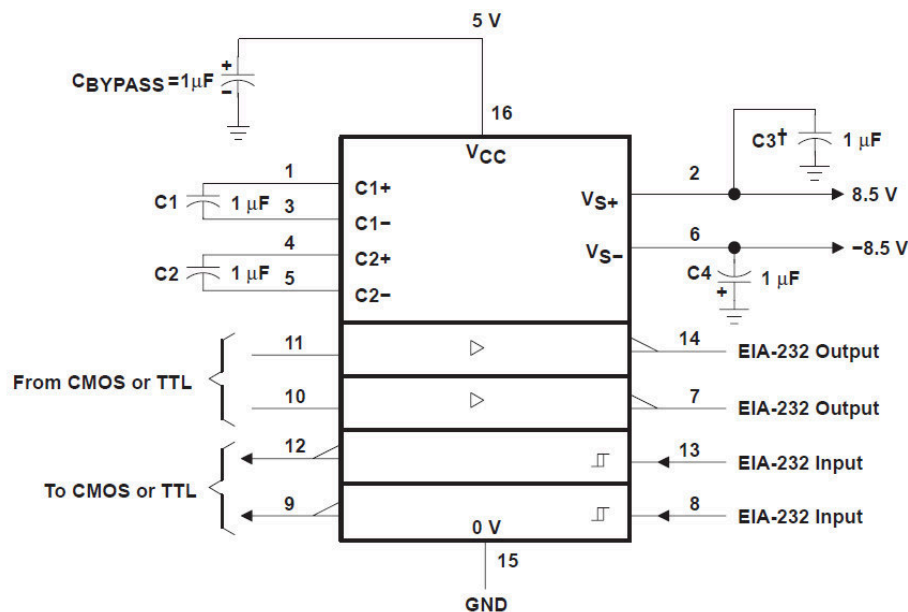
V předešlých komandech se používá slovo kurzor. Pod tímto si můžeme představit souřadnici znaku na displeji, s kterou aktuálně pracuju.

Při posílání dat a komandu je nutné i dodržovat časové rozestupy. Například nastavíme komand „reset displeje“ a musíme ho tam držet podle datasheetu minimálně 20 ms. Nebo při posílání dat (znaku) je třeba držet na vstupech displeje data po dobu cca 2 ms. Další zdržení může nastat, pokud se použije čtyřbitový mód. U tohoto módu dochází k tomu, že se nepoužívají datové vstupy DB0 až DB3, takže se posílaný komand nebo data musí poslat jako první půlka a poté druhá půlka. Toto rozseknutí sice ušetří množství potřebných výstupů z mikropočítače do displeje, ale zároveň dojde k zvětšení odezvy LCD displeje. Tato odezva není ale v naší aplikaci tak důležitá.

2.3.2 Sériová linka RS232 a převodník MAX232

RS232 je linka, která používá asynchronní sériovou komunikaci pro přenos dat. Posílaná data bývají nejčastěji 8mi bitová, ale lze posílat i 7mi nebo 9bitová slova. Jednotlivé bity se přenášejí postupně od toho s nejnižší vahou (LSB) až po ten s nejvyšší vahou (MSB). Tato linka používá dvě napěťové úrovně proti zemi. Logická 1 je záporné napětí a logická 0 je kladné napětí. Velikost tohoto napětí bývá ± 5 V, ± 10 V, ± 12 V, ± 15 V. Vždy jde tedy o napětí mezi vysílacím vodičem (TX) a zemí, nebo mezi přijímacím vodičem (RX) a zemí.

V této bakalářské práci je použita komunikace po RS232. Piny mikropočítače RX a TX (Receive a Transmit) jsou připojeny k převodníku MAX232IN. Tento obvod je převodníkem úrovní mezi TTL na RS232 a RS232 na TTL. Jeho podstatnou vlastností je, že při napájení pouze 5V si pomocí nábojové pumpy vytvoří až dvakrát větší napětí než je napětí napájení. Toto provádí pomocí elektrolytických kondenzátorů k němu připojených. Zjednodušeně řečeno je nabíje připojením k napájecímu napětí paralelně a poté je spojí sériově a tento zvětšený náboj (součet napětí více kondenzátorů do série) používá k posílání dat. TX výstup má i omezovač výstupního proudu na ± 10 mA kvůli odolnosti vůči zkratu. Z Obr. 7 Aplikačního zapojení si lze povšimnout, že MAX232 lze použít pro dva RX a dva TX vodiče (takže dvě zařízení s kterými chceme komunikovat po RS232).



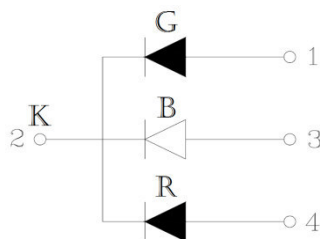
Obr. 7 Aplikační zapojení MAX232 (převzato z [5])

2.3.3 Ostatní

Mezi tyto periferie bych zařadil například část napájení. V obvodu je použit napěťový stabilizátor L7805, který je schopen dodávat trvale proud 0,75 A při stabilním výstupním napětí okolo 5 V (přesněji od min. 4,65 V do max. 5,35 V). Jeho vstup snese maximální napětí do 35 V. Vstup i výstup stabilizátoru je filtrován elektrolytickými kondenzátory. Další informace najdeme v [3].

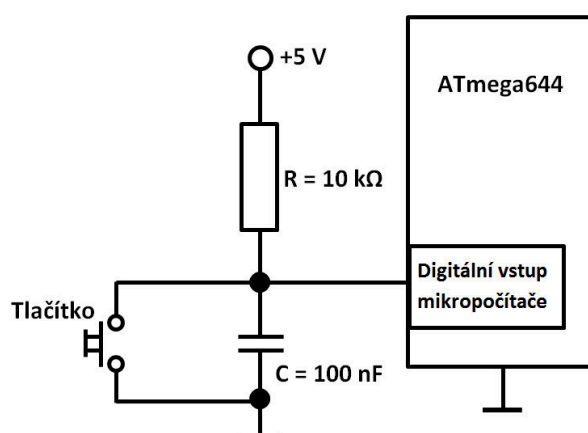
Jako další periferie bych uvedl bipolární NPN tranzistory BC549 (datasheetu v [10]), které jsou použity ve spínacím režimu jako posilovače výstupů mikropočítače ATmega. Kdyby bylo například podsvícení LCD displeje spínáno přímo přes mikropočítač, došlo by k jeho zničení, důsledkem velkého odběru proudu. Podsvícení LCD displeje si totiž podle datasheetu (v [9]) vezme proud klidně i 100 mA a pin mikropočítače může dodat proud pouze maximálně 40 mA. Zapojení tranzistorů je společný emitor.

V obvodu se nachází i RGB LED dioda, která je použita pro indikaci stavů. Tato dioda má trojbarevný čip (červený, zelený, modrý) a má čtyři vývody (Obr. 8). Jeden z vývodů je společná katoda, takže také pomocí tranzistoru, které přepínají jednotlivé nožky diody ke kladnému potenciálu napájení, dochází k rozsvěcování příslušných barev.



Obr. 8 Vnitřní zapojení RGB LED na výstupní nožky (převzato z [6])

Důležitým zapojením je i připojení tlačítek ke vstupům mikropočítače. V mém obvodu sem je zapojil s „pull-down“ rezistorem a odrušovacím kondenzátorem proti zákmitům (Obr. 9). Při stisku tlačítka se připojí vstup mikropočítače na zem (log. 0) a vybije odrušovací kondenzátor. Jakmile se tlačítko pustí, v mžiku se nabije odrušovací kondenzátor a na vstupu mikropočítače se objeví +5 V (log. 1)



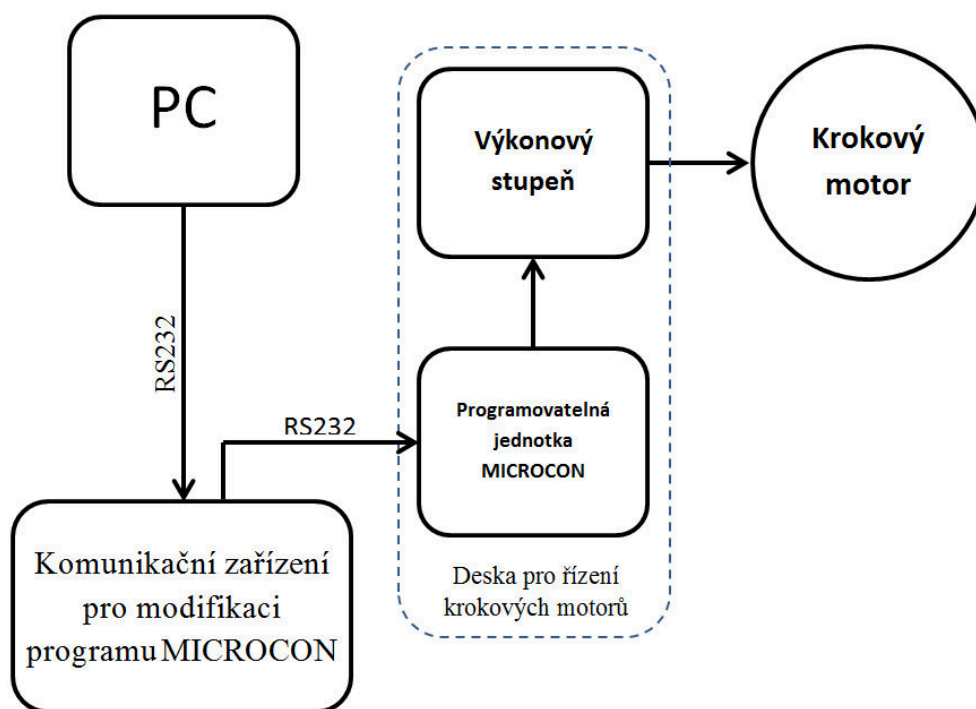
Obr. 9 Připojení tlačítka k mikropočítači

3 POPIS, POSTUP A VÝVOJ ZAPOJENÍ

V této kapitole bude popsán průběh bakalářské práce a řešené problémy, se kterými sem se setkal.

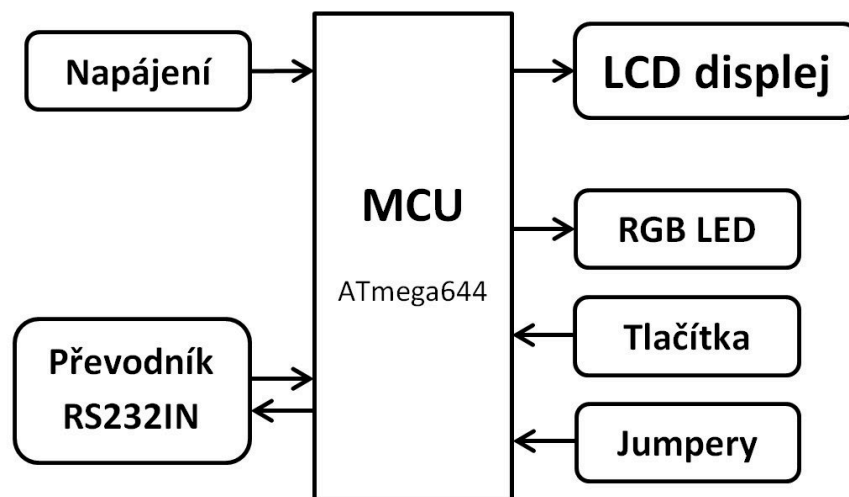
3.1 Blokové schéma zapojení

Na Obr. 10 lze vidět, kam bude umístěno zařízení této bakalářské práce. Bude spojnici mezi PC a jednotkou Microcon.



Obr. 10 Blokové schéma aplikace komunikačního zařízení pro modifikaci programu Microcon

Následný Obr. 11 blokově popisuje zapojení tohoto komunikačního zařízení. Tyto připojené periferie jsou rozebrány v předešlé kapitole.



Obr. 11 Blokové schéma zapojení komunikačního zařízení pro modifikaci programu Microcon

3.2 Popis zapojení

Na začátku bylo nutné zjistit jak jednotka Microcon komunikuje s PC přes aplikaci Inmotion PC Utilities. Pomocí programu, který umožňuje monitorovat sériový port v PC, byla „odchycena“ komunikace, zobrazená v Tabulce 1.

Tabulka 1 Komunikace po sériové lince

č. pořadí	Směr komunikace	Odeslané znaky
1.	PC→Microcon	\.^16.
2.	Microcon→PC	0
3.	PC→Microcon	U20.
4.	Microcon→PC	E
5.	PC→Microcon	U15.
6.	Microcon→PC	.
7.	PC→Microcon	[.S1000.].[.S1500.].

Rozbor jednotlivých řádků komunikace (z Tabulky 1)

Řádek č. 1. - PC posílá do Microcon jednotky (dále jen jednotka) znak lomítka, který je příkazem pro RESET jednotky (jednotka bude uvedena do výchozího stavu). Následuje znak CR (0x0D = Carriage Return = speciální netisknutelný znak, který posune kurzor na následující řádek). Význam dalších znaků: „.^16“ bohužel nebyl zjištěn. Poslední znak je opět CR.

Řádek č. 2. – Jednotka odpovídá v tomto případě znakem „0“. Tato odpověď souvisí s neznámými znaky z řádku č. 1. Při analýze více takovýchto komunikací bylo zjištěno, že se neznámé znaky z prvního řádku pokaždé mění a rovněž se mění i odpověď ve

formě jednoho znaku na řádku č. 2. Nebyla nalezena posloupnost ani vzorec této odpovědi z řádku č. 2.

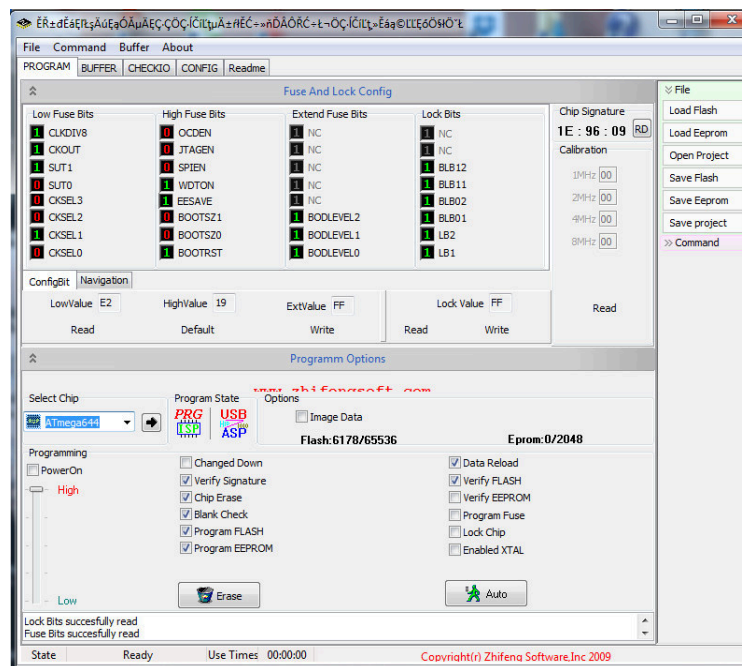
Řádek č. 3., 4., 5. a č. 6. – PC odešle „U20.“ a jemu odpovězeno jednotkou „E“. Poté PC odešle „U15.“ A je mu odpovězeno jednotkou „.“ (0x00 = speciální netisknutelný znak, který je nazýván NULL a má nulovou hodnotu). Tato sekvence je při každé komunikaci stejná. Nebylo zjištěno, co přesně tato příprava před samotným odesláním uživatelského souboru dělá. Jelikož lze přepisovat program v jednotce Microcon jen na určitých místech, tak si myslím, že se tato funkce nastavuje těmito parametry.

Řádek č. 7. – Zde je z PC odeslán samotný uživatelský povelový soubor. Tento soubor většinou začíná znakem „[“. Jednotlivé nastavované parametry jsou odděleny znakem CR (0x0D = Carriage Return = speciální netisknutelný znak, který posune kurzor na následující řádek). A Předposlední znak bývá „]“ a poslední zase CR.

Dalším krokem bylo, vyzkoušet zda půjde jednotka naprogramovat bez pomoci aplikace Inmotion PC Utilities, takže jsem se dal na návrh schématu zapojení. Zapojil jsem mikropočítač, displej, LED a tlačítka na nepájivé pole a začal psát program pro mikropočítač.

První problém nastal u samotného AVRISP programátoru mikropočítače. Koupil jsem ho na internetu z Číny a nějak jsem nečekal, že bude problém. Každé zařízení USB má svoje tzv. „USB Class“ číslo, které zařazuje zařízení podle funkce do podskupin (Např.: skupina č. 1. audio, skupina č. 2. komunikace, skupina č. 3. HMI, ...). Při vložení tohoto programátoru do USB se tvářil jako HMI zařízení (podskupina č. 3.), což neumožňovalo nainstalovat správné ovladače. Toto USB ID nelze bez přeprogramování samotného programátoru změnit. Proto jsem si musel půjčit programátor od kolegy. První problém byl vyřešen, i když nejrychlejším způsobem.

Následně pomocí programu PROGISP (Obr. 12) proběhlo připojení programátoru a mikropočítačem. V tomto programu lze nastavovat tzv. Fuse bity, což je vlastně SW nastavování HW (podobně jako Bios u PC). Musel jsem zde vypnout bity ovládající JTAG (OCDEN a JTAGEN) a zapnout děličku kmitočtu 8mi (CLKDIV8). Pak stačilo pouze vybrat typ mikropočítače v seznamu a programátor byl připraven.



Obr. 12 Prostředí programu PROGISP

Nějakou dobu trvalo, než jsem zprovoznil LCD displej. Připojil jsem totiž jeho piny k mikropočítači a nefungovalo mi nic. Po delším bádání jsem přišel na to, že mám displej připojen na piny, které mají nejen funkci vstupně/výstupních portů ale i funkci JTAG. Stačilo v programu mikropočítače, při inicializaci, nastavit bit JTD (JTag Disable = Vypnutí JTAGu) nacházející se v registru MCUCR (MCU Control Regist = registr pro nastavení mikropočítače) do log. 1. Zajímavé bylo, že musí proběhnout zápis do tohoto bitu dvakrát po sobě, jinak není zápis platný. Komunikace s displejem je 4bitová a podsvícení LCD je ovládáno pinem mikropočítače. Také je u LCD trimr, kterým je možno nastavovat kontrast displeje. Podrobnější rozebrání programu ovládající displej je v kapitole 4.

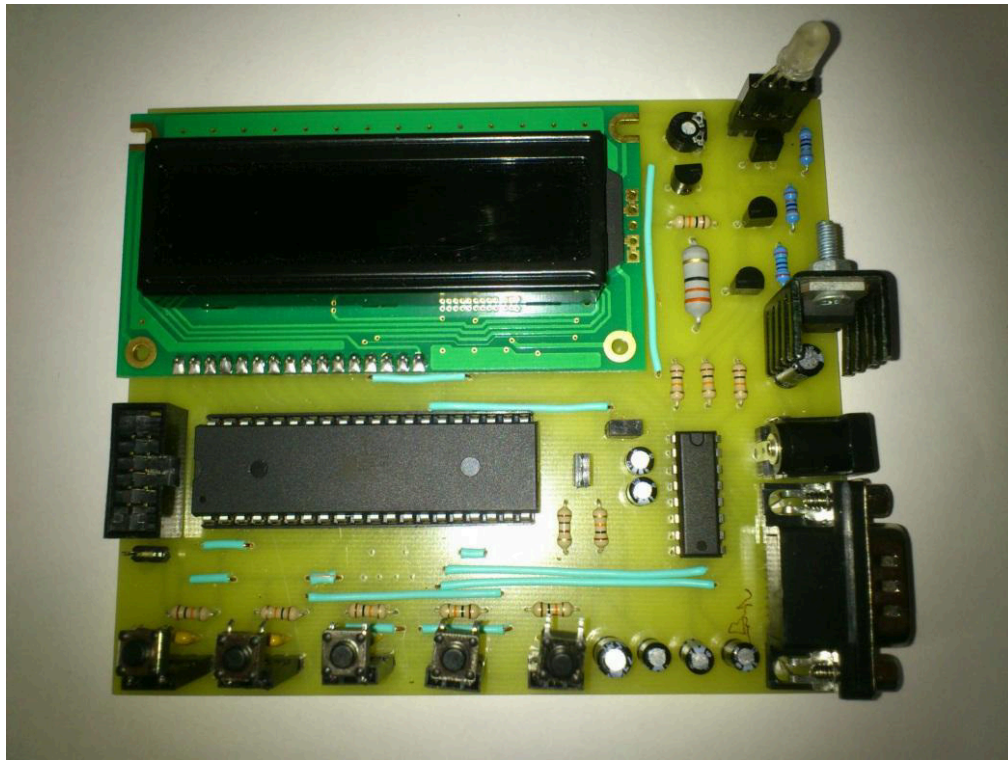
Jedním z úkolů bylo, udělat na LCD obrazovce prostředí, ve kterém se bude uživatel pohybovat pomocí tlačítek. Tlačítka mají funkci nahoru, dolů, vlevo, vpravo a OK. Při stisku tlačítka dojde v mikropočítači k externímu přerušení. Tři tlačítka jsou připojeny na klasické externí přerušení INT0, INT1 a INT2. U tohoto přerušení je třeba nastavit povolení přerušení (že má vůbec na toto přerušení mikropočítač reagovat) a nastavit, jakou hranou signálu dojde k přerušení (nástupná, sestupná, obě nebo když bude pin trvale na log. 0 tak bude mikropočítač v přerušení). Další dvě tlačítka jsou připojeny na PCINT28 a PCINT29. PCINT je tzv. přerušení od změny pinu (PCINT = Pin Change INterrupt). Toto externí přerušení může být vyvoláno od kteréhokoliv pinu mikropočítače. Před použitím této funkce je potřeba nastavit v registru PCICR (Pin Change Interrupt Control Registr = registr pro nastavení externího přerušení změny hodnoty na pinu) PORT, kterým bude přerušení vyvolávané. V našem případě jsou PCINT28 a PCINT29 na PORTu D. Tomu odpovídá, že musím nastavit v PCICR bit PCIE3 na log. 1 (PCIE je pro celý PORT D). Tímto jsme nastavili přerušení od celého PORTu. Následně musíme upřesnit, od kterého pinu přerušení bude. Takže v registru PCMSK3 nastavíme bity PCINT28 a PCINT29 do log. 1. A nastavení externích přerušení je dokončeno.

K tomu abych mohl vyzkoušet programování jednotky Microcon, bylo potřeba nastavit sériovou komunikaci USARTem (univerzální synchronně/asynchronní sériové rozhraní). Použil jsem předlohu, která byla v datasheetu mikropočítače [1]. Stačilo jen upravit nastavení USARTu podle svého požadavku. Nastavil jsem, že se bude jednat o asynchronní komunikaci, posílané data budou v 8mi bitovém formátu, bez parity a s jedním stop bitem. Definoval jsem si přenosovou rychlost na 4800 Baudů. Toto nastavení se provádělo výpočtem hodnoty UBRR, která se zapíše do časovače USARTu. Do výpočtu se zadává požadovaná přenosová rychlost a takt mikropočítače (1).

$$UBRR = \left(\frac{\text{Takt mikropočítače}}{16 \times \text{Požadovaná přenosová rychlost}} \right) - 1 \quad (1)$$

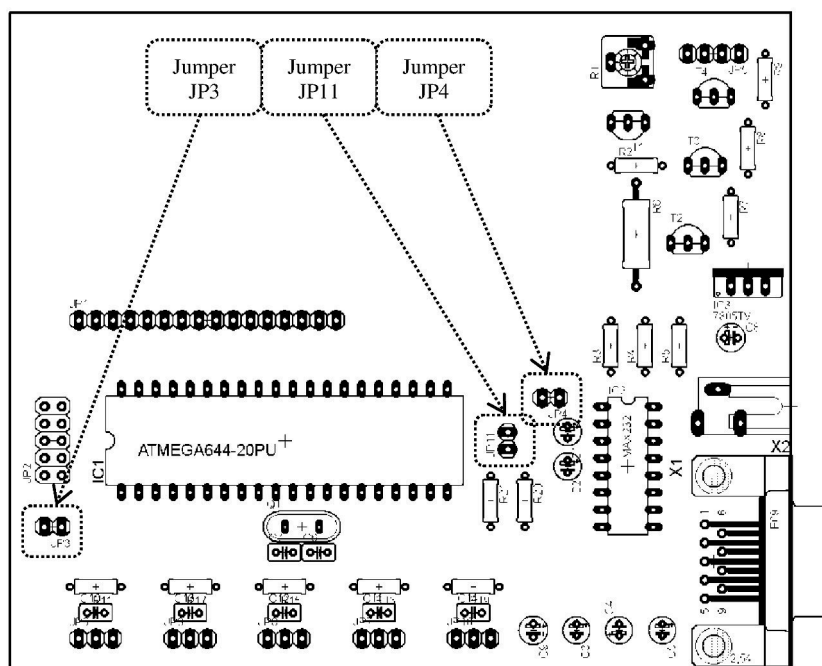
Piny mikropočítače, které mají funkci příjmu a odesílání dat USARTem jsou připojeny na převodník úrovně RS232IN. Za tímto převodníkem je už signál v úrovních linky RS232 a je vyveden na sériový konektor.

Po vyzkoušení zapojení na nepájivém poli bylo zapojení překresleno do programu Eagle a zároveň v něm byla navrhnutá i deska plošných spojů. Při kreslení schématu jsem se nesetkal s větším problémem, protože byly všechny součástky v knihovnách Eaglu. Následoval návrh samotné desky plošných spojů. Byla snaha uspořádat všechny součástky tak aby v tom nebyl chaos. Po optimálním rozmístění bohužel bylo nutné použít i propojky na vrchní straně, protože jsem zvolil, že deska bude jednovrstvá. Na desce jsou tlačítka a RGB led připojeny pomocí počítačových pinů. To umožňuje vyvést tlačítka i LED například na panel pomocí kabelů. Důležité bylo osadit i desku konektory. Je zde konektor napájení, konektor pro sériovou linku RS232 a konektor pro programátor AVRISP.



Obr. 13 Fotka hotového zařízení pro modifikaci programu Microcon

V zapojení najdeme celkem 3 tzv. jumpery. První (JP3 na Obr. 14) slouží k připojení USB napájení na desku zařízení. Lze toho využít, pokud nebude po ruce napájecí adaptér a bude potřeba mikropočítač naprogramovat. Další jumper (JP4 na Obr. 14) slouží k odpojení napájení převodníku RS232IN. Při testování na nepájivém poli se totiž stávalo, že tento převodník vyhazoval elektronickou pojistku externího napájecího zdroje. Nevěděl jsem, jestli tento problém způsobený připojenými kapacitami k obvodu RS232IN bude přetrvávat i při použití napájecího stabilizátoru L7805. Pokud by problém přetrvával, stačí místo jumperu připojit odpor, který sníží prvotní proudový impuls odebíraný převodníkem RS232IN. Poslední použitý jumper (JP11 na Obr. 14) slouží k zvolení módu. Pokud je jumper zapojen, bude zařízení pro modifikaci programu Microcon přijímat uživatelský program z PC. A když bude jumper odpojen, bude zařízení fungovat jako editor programu pro jednotky Microcon s možností odeslání uživatelského programu do jednotky Microcon.



Obr. 14 Osazovací plán součástek na DPS s vyznačenými jumpery

Odesílání uživatelského programu pro jednotku Microcon z PC se provádí pomocí aplikace Hercules SETUP utility. Bohužel nelze k tomuto použít Inmotion PC Utilities od firmy Microcon. Důvod je popsán na začátku této podkapitoly. Jedná se o problém s prvním a druhým řádkem sériové komunikace v Tabulce 1.

Postup odeslání programu Microcon z PC do mikropočítače

- Uživatel napíše program pro jednotku Microcon v Inmotion PC Utilities a celý ho označí a zkopíruje do textového editoru (Poznámkový blok).
- V textovém editoru naskládá všechny příkazy do prvního řádku a oddělí je mezerou.
- Upravený text zkopíruje a pošle programem Hercules SETUP utility

Tento postup odeslání bude funkční, pokud bude zapojen jumper JP11, kterým se vybírá mód. Kdyby byl jumper JP11 odpojen, nebude program do mikropočítače přenesen.

Po stisku tlačítka vlevo, vpravo, nahoru nebo dolů se zobrazí na displeji LCD uživatelské rozhraní. V tomto rozhraní může uživatel upravovat hodnoty jednotlivých parametrů a v závorce za nimi vidí hodnotu před úpravou.

Pokud bude uživatel chtít upravený program odeslat do jednotky Microcon, musí být nezapojený jumper JP11 a jednotka připojena ke konektoru RS232.

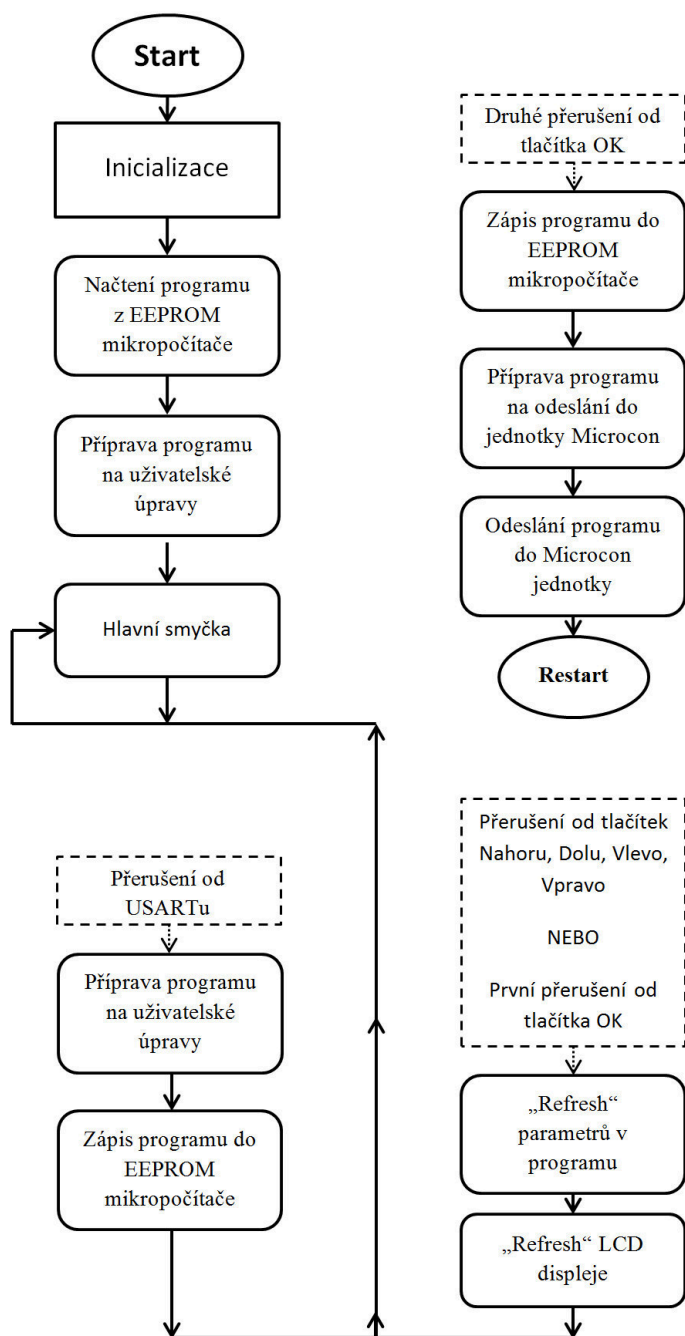
Po odeslání programu se upravený program uloží do EEPROM paměti mikropočítače a provede se reset mikropočítače. Následně proběhne znovu inicializace a bude načten Microcon program z EEPROM paměti mikropočítače. Komunikační zařízení pro modifikaci programu Microcon je znova připraveno na příjem jiného programu z PC nebo na odesílání modifikovaného programu do jednotky Microcon. Výběr funkce je řízen již zmíněným jumperem JP11.

Celou dobu práce se zařízením bude RGB LED indikovat stav mikropočítače. Pokud bude svítit zeleně, je mikropočítač připraven na interakce uživatele. Jestliže bude LED svítit červeně, probíhají operace s poli a mikropočítač je zaneprázdněn. A v poslední řadě, pokud bude LED svítit nebo blikat modře, mikropočítač odesílá nebo obdrží data z RS232.

4 POPIS PROGRAMU MIKROPOČÍTAČE

Největší část práce probíhala v aplikaci Atmel studio. Vytvářel jsem zde program pro mikropočítač. V této kapitole bude popsána hlavní smyčka programu a popsán příklad průběhu programu.

4.1 Blokový diagram programu



Obr. 15 Blokový diagram programu

Při zapnutí mikropočítače se provede prvně inicializace. Tento podprogram slouží k prvotnímu nastavení všech registrů a periférii. Poté se načte program pro řídicí jednotku Microcon z EEPROM paměti mikropočítače a provede se jeho příprava na úpravy uživatelem. V klidovém stavu se v hlavní smyčce nic zajímavého neděje. Program je zde zacyklen. Další fungování programu je ovládáno přerušeními.

Přerušení od USARTu

Pokud přijde přerušení od USARTu tak se přijatá informace uloží do dočasného pole. V programu je udělaný podprogram, který má funkci „timeout“ (= vypršení času) příjmu USARTu. Při posílání Microcon programu nastává problém v tom, že není tento program ukončen nějakým ukončovacím znakem. Proto je použit zmíněný časovač pro USART, který tento problém řeší. Pokud totiž nedojde před uplynutím dvou jeho přetečení k dalšímu příjmu USARTem, program tento stav označí jako dokončený příjem.

Následuje identifikace dat v poli co USART přijmul. To je ovlivněno jumperem JP11. Pokud bude zapojen, jde o přijatý program z PC. Pokud bude odpojen, jde o odpověď od jednotky Microcon při jejím programování. Jsou-li data identifikována jako program z PC, dojde k jeho připravení na úpravy uživatelem a uloží se do EEPROM paměti mikropočítače.

Přerušení od tlačítek nahoru, dolů, vlevo, vpravo

Přerušení od těchto tlačítek provádí aktualizaci displeje a úpravu samotných hodnot programu Microcon. Uživatel vidí upravovanou hodnotu, kterou zvyšuje nebo snižuje. V závorce za ní je původní hodnota povelu.

Přerušení od prvního stisku tlačítka OK

Při tomto přerušení bude uživatel dotázán, zda chce upravené parametry do jednotky Microcon opravdu odeslat.

Přerušení od druhého stisku tlačítka OK

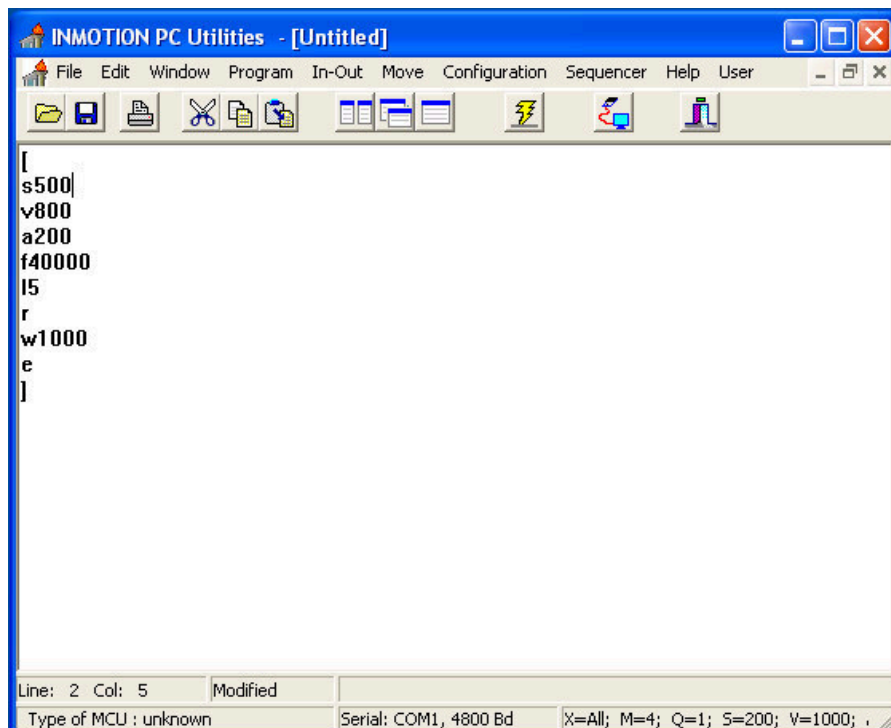
Tímto přerušením je zahájena sekvence odesílání upraveného programu Microcon do jednotky Microcon. Po odeslání se program uloží do EEPROM paměti mikropočítače a provede se RESTART mikropočítače.

4.2 Příklad modifikace programu Microcon

V této podkapitole bude detailně rozebrán postup práce se zařízením pro modifikaci programu Microcon. Tento příklad je rozdělen na část, když posíláme program Microcon z PC do mikropočítače a část když posíláme Microcon program z mikropočítače do jednotky Microcon.

4.2.1 Odeslání programu Microcon z PC do mikropočítače

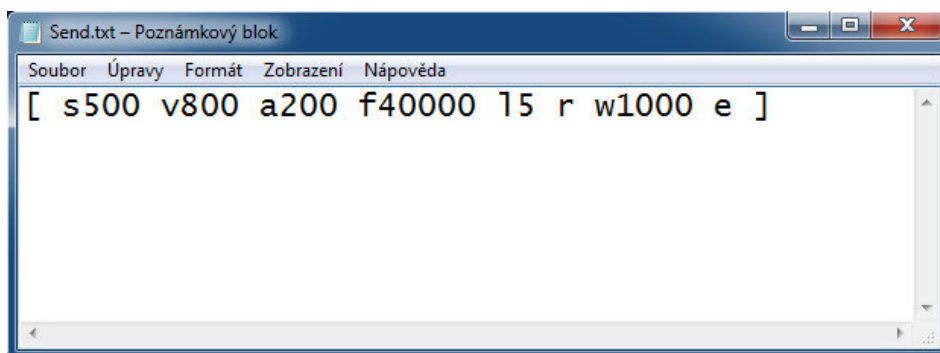
Uživatel napíše program pro jednotku Microcon v programu Inmotion PC Utilities



Obr. 16 Vytvoření programu pro jednotku Microcon v Inmotion PC Utilities

Úprava programu v textovém editoru

Jak bylo dříve zmíněno, musí být vše v jednom řádku a odděleno mezerou.

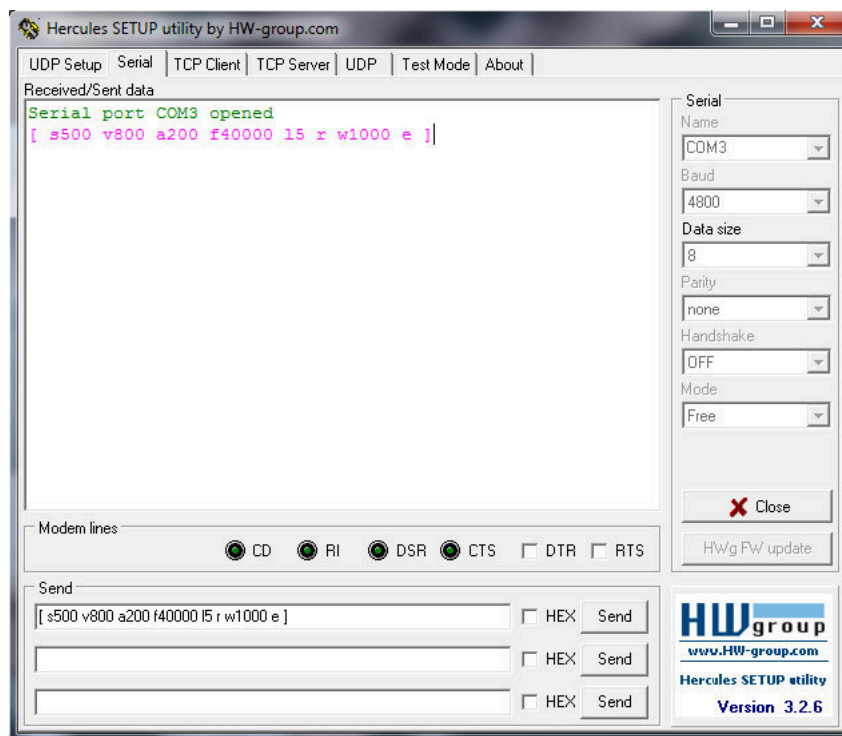


Obr. 17 Program pro Microcon v textovém editoru

Odeslání Microcon programu do mikropočítače

Toto odeslání probíhá pomocí aplikace Hercules SETUP utility (Obr. 18). Tato aplikace je volně stažitelná [7]. V záložce „Serial“ nastavíme komunikační parametry (číslo portu, rychlost, paritu, stop bity a velikost dat) a port otevřeme. Následně zkopírujeme do prvního

řádku „Send“, ve spodní části obrazovky, upravený program pro Microcon z textového editoru z předešlého kroku. Pokud již tak máme, stačí odeslat tlačítkem „Send“ příslušný řádek. Před odesláním se uživatel ujistí, jestli je zapojen jumper JP11, který nastavuje, že přijatá data mikropočítačem jsou nový program Microcon (pokud bude tento jumper JP11 odpojen, neproběhne příjem programu do mikropočítače správně).



Obr. 18 Aplikace Hercules SETUP utility

Zpracování přijatých dat mikropočítačem

V následující tabulce (Tabulka 2) je napsáno, co mikropočítač přijmul. Znaký jsou napsané jak v hexadecimální podobě tak i podle ANSI tabulky.

Tabulka 2 Přijaté data mikropočítačem

Pořadí znaku	0.	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
Znak v hexadecimální soustavě	5B	20	73	35	30	30	20	76	38	30	30
Znak v ASCII	[„mezera“	s	5	0	0	„mezera“	v	8	0	0

Pořadí znaku	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.
Znak v hexadecimální soustavě	20	61	32	30	30	20	76	34	30	30
Znak v ASCII	„mezera“	a	2	0	0	„mezera“	f	4	0	0

Pořadí znaku	21.	22.	23.	24.	25.	26.	27.	28.	29.
Znak v hexadecimální soustavě	30	30	20	6C	35	20	72	20	77
Znak v ASCII	0	0	„mezera“	1	5	„mezera“	r	„mezera“	w

Pořadí znaku	30.	31.	32.	33.	34.	35.	36.	37.
Znak v hexadecimální soustavě	31	30	30	30	20	65	20	5D
Znak v ASCII	1	0	0	0	„mezera“	e	„mezera“]

Následně proběhne rozřídění přijatých znaků do dvourozměrného pole MicroconProg. Toto třídění probíhá tak, že se bere znak po znaku (od 0. po poslední) a ukládá se každý další znak na další sloupec vpravo. Při této operaci se kontroluje, zda se znak nerovná „mezeře“ (hexadecimálně = 20) a pokud ano, pokračuje zápis do pole na dalším řádku (o jeden níže). Toto pole bude vypadat v tomto případě jako v Tabulce 3.

Tabulka 3 Dvourozměrné pole rozříděných znaků MicroconProg

	0	1	2	3	4	5	6
0	[
1	s	5	0	0			
2	v	8	0	0			
3	a	2	0	0			
4	f	4	0	0	0	0	
5	l	5					
6	r						
7	w	1	0	0	0		
8	e						
9]						

Tímto krokem jsme dostali program Microcon do stavu jako v PC v aplikaci Inmotion PC Utilities. Následuje krok, ve kterém definujeme, které parametry bude moci uživatel na LCD měnit. V programu pro mikropočítač je podprogram „EditovaneParametry“ na Obr. 19.

```
void EditovaneParametry(){
    ProgSeparation('v');
    ProgSeparation('f');

};
```

Obr. 19 Podprogram „EditovaneParametry“

V tomto podprogramu se volá metoda „ProgSeparation“ s parametrem povelu programu Microcon (povelem programu Microcon je myšleno např.: w, v, f, a, ...), který bude měnitelný uživatelem. V našem případě bude mít možnost uživatel měnit velikost maximální rychlosti „v“ a velikost dráhy pohybu vpřed „f“.

Metoda (podprogram) „ProgSeparation“ porovnává každý řádek v nultém sloupci pole MicroconProg (Tabulka 3) s volaným znakem (v našem případě „v“ a „f“). Pokud najde shodu, uloží si jeho adresu a znak, který porovnával. Metoda bude prvně volána znakem „v“, poté znakem „f“.

Následně je volán podprogram „GUIPreparation“. Tento podprogram se stará o vykreslování znaků na LCD displej. Obsahuje metody pro posun kurzoru, zobrazení výběru upravitelných parametrů programu Microcon, číselnou hodnotu aktuálně vybraného upravovaného parametru programu Microcon a číselnou hodnotu aktuálně vybraného upravovaného parametru programu Microcon před změnou uživatelem. Prostředí LCD displeje vypadá v našem případě jako na Obr. 20.



Obr. 20 Prostředí LCD displeje

V prvním řádku se dá pohybovat pomocí tlačítek doleva a doprava. Provádí se zde výběr upravovaného povelu programu Microcon. Za znakem povelu je vždy číslo. Program Microcon může totiž obsahovat více stejných povelů s jiným nebo i stejným parametrem (např.: v200, v500, v300, v200). Dříve zmíněná metoda ProgSeparation přiřazuje číslo ke znaku v pořadí odshora dolu v poli MicroconProg (např.: v0, v1, v2, v3). Na Obr. 20 je aktuálně vybrán parametr „v0“. Pod tímto parametrem, v druhém řádku, je hodnota, která už byla upravena uživatelem „820“. Vpravo, v kulaté závorce, je původní hodnota aktuálního parametru. Hodnota aktuálně vybraného parametru se mění pomocí tlačítek nahoru a dolů. Tyto změny hodnoty jsou přepisovány přímo do pole MicroconProg.

Setkal jsem se zde s problémem LCD displeje. Po displeji mohu chtít zobrazovat znaky ale ne například datový typ integer. Takže upravovaná číselná hodnota ve formě znaků v poli MicroconProg musela být převedena pomocí speciální funkce na integer. Tento integer je upravován stiskem tlačítek nahoru a dolů. Při každém stisku tlačítek (nahoru a dolů) je zpátky převáděn na znaky, které přepisují minulou hodnotu v poli MicroconProg.

4.2.2 Odeslání programu Microcon mikropočítačem do jednotky Microcon

V předchozí podkapitole jsme umožnili uživateli upravovat parametry programu Microcon a v této podkapitole bude popis odeslání do jednotky Microcon.

Pokud bude chtít uživatel upravený program Microcon odeslat do jednotky Microcon zmáčkne tlačítko „OK“. Na LCD displeji se mu zobrazí, zda chce opravdu upravený program Microcon odeslat do jednotky. Když zmáčkne tlačítko „OK“ podruhé, zahájí se příprava programu před odesláním a sekvence odesílání.

Příprava programu před odesláním do jednotky Microcon

Před samotným odesláním je nutné data z pole MicroconProg „naskládat“ za sebe do jednorozměrného pole „TXMicroconProg“. Tuto funkci plní podprogram „ProgTXPreparation“. ProgTXPreparation funguje tak, že se pohybuje v každém řádku pole MicroconProg a ukládá každou hodnotu do pole jednorozměrného. Když je buňka v řádku prázdná, zapíše do jednorozměrného pole znak CR (0x0D = Carriage Return = speciální netisknutelný znak) a pokračuje od začátku následujícího řádku pole MicroconProg. Znak CR se zde zapisuje, protože to vyplývá z analýzy komunikace aplikace Inmotion PC Utilities a jednotky Microcon (podkapitola 3.2) V našem případě bude pole TXMicroconProg vypadat jako v Tabulce 4.

Tabulka 4. Jednorozměrné pole TXMicroconProg

Pořadí znaku	0.	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
Znak v hexadecimální soustavě	5B	0D	73	35	30	30	0D	76	38	32	30	0D	61
Znak v ASCII	[CR	s	5	0	0	CR	v	8	2	0	CR	a

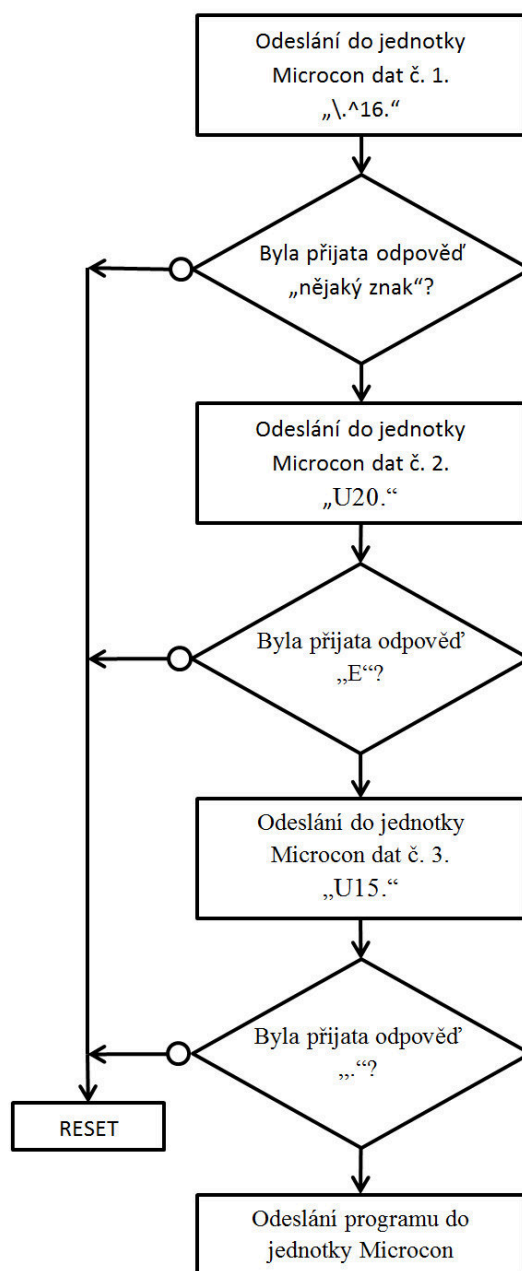
Pořadí znaku	13.	14.	15.	16.	17.	18.	19.	20.	21.	22.	23.	24.	25.
Znak v hexadecimální soustavě	32	30	30	0D	76	34	30	30	30	30	0D	6C	35
Znak v ASCII	2	0	0	CR	f	4	0	0	0	0	CR	l	5

Pořadí znaku	26.	27.	28.	29.	30.	31.	32.	33.	34.	35.	36.	37.
Znak v hexadecimální soustavě	0D	72	0D	77	31	30	30	30	0D	65	0D	5D
Znak v ASCII	CR	r	CR	w	l	0	0	0	CR	e	CR]

V tomto stavu je pole TXMicroconProg připraveno k odeslání do jednotky Microcon.

Sekvence odeslání programu do jednotky Microcon

Z analýzy komunikace (podkapitola 3.2), mezi programem Inmotion PC Utilities jednotkou Microcon bylo zjištěno, že program Inmotion odesílá inicializační data a jednotka Microcon mu odpovídá. To bylo vyřešeno sekvencí zobrazené na Obr. 21.



Obr. 21 Sekvence odesílání programu do jednotky Microcon

Při odesílání dat do jednotky Microcon je nutné dělat mezi jednotlivými znaky 0,5ti sekundové mezery. Podle datasheetu jednotky Microcon je toto doporučený čas. Důvodem je, že jednotka Microcon obsahuje paměť EEPROM s dlouhou dobou zápisu.

Po odeslání programu do jednotky Microcon se pole MicroconProg uloží i do EEPROM paměti mikropočítače. Následně je proveden SW reset mikropočítače. Při tomto resetu se mikropočítač znovu inicializuje a načte si pole MicroconProg z paměti EEPROM. Uživatel nemusí znovu posílat z PC program Microcon, pokud teda nechce změnit celou jeho strukturu. Zápis do paměti EEPROM se provádí pomocí metod z datasheetu

4.3 Popis některých podprogramů

Obsluha LCD displeje

Knihovna pro ovládání a obsluhu LCD displeje byla nalezena na internetu [8]. Knihovnu jsem si upravil a přizpůsobil. Využil jsem z ní: `LCD_Init()`, `LCD_Set_Cursor()`, `LCD_Write_Char()` a `LCD_Write_String()`.

LCD_Init()

Pomocí tohoto podprogramu se inicializuje displej. Pošle se do něj příkaz na reset, že bude fungovat v 4bitovém režimu, vypne se zobrazení kurzoru a jeho blikání. Jednotlivé nastavovací příkazy jsou odesílány s určitým časovým zpožděním. Tento podprogram se zavolá vždy pouze jednou a to při zapnutí mikropočítače a napájení LCD displeje.

LCD_Write_Char() a LCD_Write_String

Tyto dvě funkce zapisují na předem nastavenou adresu kurzorem (`LCD_Set_Cursor()`) znak (char) nebo string. Správná syntaxe:

- `LCD_Write_Char('Z')` – ASCII zadání znaku zapíše na nastavenou adresu znak „Z“
- `LCD_Write_Char(0x30)` – hexadecimální zadání zapíše na nastavenou adresu znak „0“
- `LCD_Write_String("Ahoj svete")` – Zapíše od nastavené adresy „Ahoj svete“

LCD_Set_Cursor()

Tímto podprogramem nastavujeme adresu na displeji, od které bude ukládán string nebo na kterou se uloží znak. Nastavení probíhá takto: `LCD_Set_Cursor` (řádek (1 nebo 2), sloupec (0 až 15)). Na Obr. 22 je příklad uložení dvou stringů na LCD displej.

```
LCD_Set_Cursor(1,0);  
LCD_Write_String("Bc. MicroconEdit");  
LCD_Set_Cursor(2,0);  
LCD_Write_String("PWRDby ATmega644");
```

Obr. 22 Příklad uložení stringu na LCD displej

USART

Pokud budeme chtít odeslat USARTem znak, napíšeme příkaz jako na Obr. 23 (hexadecimální podoba znaku) nebo v dvojitéch uvozovkách (např.: „1“). Po zavolání tohoto podprogramu se vstupním parametrem (znak, který chceme odeslat), se podprogram sám postará o odeslání zadaného znaku.

```
USART0_Transmit(0x31);
```

Obr. 23 Odesílání znaku „0x31“

5 ZÁVĚR

Při tvorbě této bakalářské práce jsem nabyl mnoha nových zkušeností. Musel jsem vymyslet zapojení elektronických součástek, navrhnout desku plošných spojů a především programovat mikropočítač v jazyce C. Narazil jsem na velké množství problémů, kterými byla například analýza sériové komunikace, což znamenalo zkoušet mnoho programů monitorující sériový port. Nakonec jsem našel volně šiřitelnou aplikaci Hercules SETUP utilities. Také bylo potřeba vyřešit problém s programátorem AVRISP, který nefungoval. A v neposlední řadě byly výsledkem těchto problémů desítky hodin programování v jazyce C, což považuji za největší zkušenost. Dříve jsem na takto obsáhlé úrovni neprogramoval, takže mi nezbývalo nic jiného, než se vše naučit anebo to sám vymyslet.

K samotné funkčnosti zařízení mohu říct, že funguje. Otestoval jsem modifikování programu s jednotkou Microcon s připojeným krokovým motorem. Testy proběhly úspěšně.

Jako vylepšení této bakalářské práce do budoucna by bylo použití součástek v SMD pouzdru, tím by se zmenšila velikost desky plošných spojů. Tyto součástky by musely být zároveň dimenzovány teplotně, a to ve větším rozsahu, protože v provozu se u vstříkovacích lisů vyskytují vysoké teploty. K dalšímu vylepšení by došlo, pokud by byl použit mikropočítač se dvěma USARTy (nemuselo by se řešit přepínání módu pomocí jumperu JP11).

V neposlední řadě bych vytvořil aplikaci pro PC, která by odesílala program Microcon do mikropočítače automaticky (nebo alespoň jednodušeji). Vytvoření takovéto aplikace by mohlo probíhat v prostředí programu MATLAB, který používáme ve firmě.

K dalším vylepšením HW a SW by došlo po testovacím provozu, který by trval několik týdnů. Přece jen nedokážu doma napodobit podmínky provozu, kde je zařízení pro modifikaci programu Microcon použito.

LITERATURA

- [1] Atmel. *ATmega644 datasheet* [online]. 6/2005, poslední revize 2/2012 [cit. 10.4.2014]. Dostupné z: <<http://www.atmel.com/Images/doc2593.pdf>>.
- [2] Microcon. *Manuál kontroléru M1486* [CD-ROM]. 7.6.2002, poslední revize 11/2009 [cit. 14.4.2014]. (Firma Microcon dala písemné potvrzení o možnosti kopírování tohoto materiálu do bakalářské práce.)
- [3] Thomson. *7805 datasheet* [online]. 11/2004, poslední revize 9. listopadu 2012 [cit. 12.4.2014]. Dostupné z: <<http://www.gme.cz/img/cache/doc/330/149/7805-stm-datasheet-1.pdf>>.
- [4] Microcon. *Manuál desky CD30M* [CD-ROM]. 2001 [cit. 13.4.2014]. (Firma Microcon dala písemné potvrzení o možnosti kopírování tohoto materiálu do bakalářské práce.)
- [5] Texas Instruments. MAX232IN. [online], 2/1989, poslední revize 3/2004 [cit. 10.4.2014]. Dostupné na www: <<http://www.gme.cz/img/cache/doc/329/036/max232in-datasheet-1.pdf>>.
- [6] iB. *RGB LED 599R2GBC-CC datasheet* [online]. 19.10.2005, [cit. 18.4.2014]. Dostupné na www: <<http://www.gme.cz/img/cache/doc/511/846/led-5mm-rgb-cc-1000-100-datasheet-1.pdf>>
- [7] HW group, *Hercules SETUP utility* [počítačový program]. Ver. 3.2.6. [Česká republika], [cit. 20.4.2014]. Dostupné z: <http://www.hw-group.com/products/hercules/index_cz.html> obchod@HWg.cz. Vyžaduje Windows 2000 a vyšší.
- [8] electroSome. *Interfacing LCD with Atmega32 Microcontroller using Atmel Studio* [online], 5/2013 [cit. 19.4.2014]. Dostupné na www: <<http://electrosome.com/interfacing-lcd-atmega32-microcontroller-atmel-studio/>>
- [9] WINSTAR. *Alfanumerický LCD displej wh1602a-tti-et datasheet* [online] [cit. 25.4.2014]. Dostupné na www: <<http://www.gme.cz/img/cache/doc/513/220/lcd-alfanumericky-displej-winstar-wh1602a-tti-et-datasheet-1.pdf>>
- [10] Semiconductor Components Industries. *BC549B/D datasheet* [online]. 2/2001 [cit. 22.4.2014]. Dostupné z: <<http://www.gme.cz/img/cache/doc/210/031/bc549b-datasheet-1.pdf>>.

SEZNAM PŘÍLOH

A) Schéma zapojení	1
B) Přehled povelu M1486	1
C) Schéma zapojení (Eagle)	CD-ROM
D) Deska Plošných Spojů (Eagle)	CD-ROM
E) Manuál kontroléru M1486	CD-ROM
F) Manuál desky CD30M	CD-ROM

B) Přehled povelů M1486

POVEL	POPIS
\	"Reset" uvedení kontroleru do výchozího stavu
@ (Num)	"Address" zadané číslo návěští se přiřadí této programové řádce
A (Num)	"Acceleration" zrychlení, rozsah = 1 až 65 000 kroků/s ²
B (Num)	"Backward" zpět, zadání dráhy v negativním směru
C (Num 1 to 21)	"Clear" nastavení zadaného výstupu do hodnoty logická nula
C (Num 40 to 65)	"Clear" potlačení zadané přídavné funkce
C75	"Clear Kill" obnovení vykonávání programu
C (Num 80 to 87)	"Clear" nastavení zadaného uživatelského příznaku na hodnotu OFF
C100	"Clear" vynulování čítače absolutní polohy
D	"Direction" směr, změna směru příštího pohybu
E	"End of loop" konec smyčky
F (Num)	"Forward" dopředu, zadání dráhy v pozitivním směru
G (Num)	"Go absolute" dráha zadána absolutní polohou
G+	"Go positive" trvalý pohyb v pozitivním směru až do externího přerušení
G-	"Go negative" trvalý pohyb v negativním směru až do externího přerušení
H	"Home" vykonaj pohyb do výchozí polohy
I (Num) (Value) (Num)	"If" skok na zadané návěští, jestliže zadaný vstup odpovídá zadané hodnotě ("H" High - logická jedna, "L" Low - logická nula)
J (Num)	"Jump" skok na zadané návěští
K	"Kill" okamžitý přechod do brzděného režimu, přerušení provádění povelového souboru
L (Num)	"Loop" smyčka, opakuj provádění následujících instrukcí
M (Num)	"Microstepping" počet mikrokroků na celokrok v dolním pásmu rychlosti
N (Num)	"Number" výběr tvaru průběhu proudu při mikrokrokování
O (Num)	"One" čekej dokud zadaný vstup nebude mít hodnotu logická jedna
P (Num)	"Profile" nastavení rychlosti, při které se lineární rozběhová charakteristika mění na parabolickou
Q (Num)	"Qualification" počet mikrokroků na celokrok v horním pásmu rychlosti
R	"Run" vykonaj pohyb s aktuálními hodnotami parametrů
S (Num)	"Start/stop" rychlost start/stop; rozsah = 16 až 1 950 kroků/s
T (Num 1 to 21)	"Turn on" nastavení zadaného výstupu do hodnoty logická jedna
T (Num 40 to 65)	"Turn on" zapnutí zadané přídavné funkce
T (Num 80 to 87)	"Turn on" nastavení zadaného uživatelského příznaku na hodnotu ON
U (Num)	"Upload" vyšli hodnotu čítače absolutní polohy, interní proměnné, uživatelských příznaků či přídavných funkcí
V (Num)	"Velocity" zadání maximální rychlosti
W (Num)	"Wait" čekej zadaný počet milisekund; rozsah = 1 až 16 000 000
X (Num)	"index" volba kontroleru
Z (Num)	"Zero" čekej dokud zadaný vstup nebude mít hodnotu logická nula
["Disable" odklad provedení následujících povelů
]	"Enable" provedení předchozích povelů
((Num)	"Seek negative" jdi na limit v negativním směru
) (Num)	"Seek positive" jdi na limit v pozitivním směru
= (Num)	"Equal" přiřazení zadané hodnoty čítači absolutní polohy
: (Num)	Ulož zadanou hodnotu do interní proměnné
? (Num)	Načti data na specifikovaných vstupech a ulož do interní proměnné
! (Num)	Zapiš hodnotu interní proměnné na specifikované výstupy
+ (Num)	Přičti zadanou hodnotu k interní proměnné
- (Num)	Odečti zadanou hodnotu od interní proměnné
/ (Num)	Dělení interní proměnné zadanou hodnotou
* (Num)	Násobení interní proměnné zadanou hodnotou
> (Num)	Přesun dat z interní proměnné do zadaného registru
< (Num)	Přesun dat ze zadaného registru do interní proměnné
' (Num)	Skok na podprogram začínající na zadaném návěští
.	Konec podprogramu